(2023 Spring) CS 154/454 – Python Programming II
Instructor: Poom Pianpak
Assignment 4

Description

Many people, especially youths, have complained that their lives are more difficult than older generations. Is this just a mere conjecture or does it have a statistical merit? In this assignment, we will try to answer this question by analyzing the data to determine how difficult or easy it is to afford a house between any two years.

This assignment is designed for you to develop just one big program named nm.py. The program loads (i.e., reads) 3 datasets: (1) Median Household Income in New Mexico; (2) All-Transactions House Price Index for New Mexico; and (3) Consumer Price Index: All Items for the United States. It takes 2 inputs from the user: (1) base year; and (2) target year. Then, it produces a line chart that shows rates of change of the data (in the 3 datasets) in each year when compared to the base year. It also shows quantitatively, in the chart, how much harder/easier it is to afford a house in the target year when compared to the base year.

Your submission should only contain nm.py, and please **do not** compress it into a zip file. Please name the files according to the instructions. Do not put anything else such as your name or ID as part of the file names. You also do not need to include the datasets and any generated files.

There are 3 parts to the program. You need to implement them sequentially.

(70 points) Part 1:

1. Download the 3 datasets with the maximum range of data available:
   a) **M**edian **H**ousehold **I**ncome in New Mexico (**MHI**). https://fred.stlouisfed.org/series/MEHOINUSNMA646N
   b) All-Transactions **H**ouse **P**rice **I**ndex for New Mexico (**HPI**). https://fred.stlouisfed.org/series/NMSTHPI
   c) **C**onsumer **P**rice **I**ndex: All Items for the United States (**CPI**). https://fred.stlouisfed.org/series/USACPIALLMINMEI
2. Produce a DataFrame by merging the datasets MHI and HPI. The requirements are as follows:
   a) The DataFrame consists of one level of index named YEAR, and 4 columns, all in the same level, named MHI, HPI, D_MHI, R_HPI, respectively. Note: Do not confuse the datasets MHI and HPI with the columns MHI and HPI.
   b) The datasets MHI and HPI are merged in such a way that if any of them have data at any year, the data (with their corresponding years) must exist in the DataFrame. For example, if MHI has no data at the year 1980, but HPI has 123 (assumed), then the year 1980 must be in the dataFrame with NaN and 123 as values in the columns MHI and HPI, respectively.
   c) The years (as the index) are in ascending order.
   d) There is no gap between two consecutive years in the index. That is, if the years 1999 and 2001 are the index, the year 2000 must also be there. The downloaded datasets should not have any missing year, but your program **must** still be able to fill in any missing year. Every missing year (that is filled) must fill NaN in its corresponding columns (all 4 of them). Hint: Test your program by removing some years in both the MHI and HPI datasets.
   e) The column MHI contains average values within each year in the MHI dataset.
   f) The column HPI contains average values within each year in the HPI dataset.
   g) The column D_MHI contains differences between the current and a prior element in the column MHI. Use the NaN value if there is no prior element or the prior element is NaN.
   h) The column R_HPI contains percentage changes between the current and a prior element in the column HPI. Use the NaN value if there is no prior element or the prior element is NaN.
3. **Print** the DataFrame and **save** it as a file named “nm.csv”.

Example:

Suppose that the years 1990, 2000, and 2010 are missing from both the MHI and HPI datasets, then the content of the DataFrame that your program produces should be as follows:

YEAR	MHI	HPI	D_MHI	R_HPI
1975	NaN	57.7450	NaN	NaN
1976	NaN	62.6750	NaN	0.085375
1977	NaN	72.4125	NaN	0.155365
1978	NaN	80.0975	NaN	0.106128
1979	NaN	92.5650	NaN	0.155654
1980	NaN	NaN	NaN	NaN
1981	NaN	111.0875	NaN	NaN
1982	NaN	118.9575	NaN	0.070845
1983	NaN	124.7600	NaN	0.048778
1984	20630.0	124.3100	NaN	-0.003607
1985	20423.0	127.3250	-207.0	0.024254
1986	19845.0	132.0550	-578.0	0.037149
1987	20758.0	132.8425	913.0	0.005963
1988	19296.0	130.5100	-1462.0	-0.017558
1989	22602.0	132.4425	3306.0	0.014807
1990	NaN	NaN	NaN	NaN
1991	26540.0	138.1225	NaN	NaN
1992	25860.0	145.6025	-680.0	0.054155
1993	26758.0	154.8175	898.0	0.063289
1994	26905.0	170.5025	147.0	0.101313
1995	25991.0	182.2050	-914.0	0.068635
1996	25086.0	187.0575	-905.0	0.026632
1997	30086.0	190.7500	5000.0	0.019740
1998	31543.0	195.3475	1457.0	0.024102
1999	32574.0	197.6650	1031.0	0.011863
2000	NaN	NaN	NaN	NaN
2001	33124.0	208.6725	NaN	NaN
2002	35457.0	216.3075	2333.0	0.036588
2003	35105.0	227.3525	-352.0	0.051062
2004	39562.0	243.7475	4457.0	0.072113
2005	38947.0	272.7500	-615.0	0.118986
2006	40028.0	309.0550	1081.0	0.133107
2007	44356.0	328.4575	4328.0	0.062780
2008	42102.0	324.4300	-2254.0	-0.012262
2009	43542.0	311.5875	1440.0	-0.039585
2010	NaN	299.8875	NaN	-0.037550
2011	41982.0	286.4575	NaN	-0.044783
2012	43424.0	281.2625	1442.0	-0.018135
2013	40166.0	283.4500	-3258.0	0.007777
2014	46686.0	285.9500	6520.0	0.008820
2015	45119.0	291.6350	-1567.0	0.019881
2016	48451.0	300.1500	3332.0	0.029197
2017	45601.0	309.4825	-2850.0	0.031093
2018	48283.0	321.0625	2682.0	0.037417
2019	53113.0	336.9275	4830.0	0.049414
2020	50906.0	357.5375	-2207.0	0.061170
2021	53463.0	404.7725	2557.0	0.132112
2022	NaN	469.7475	NaN	0.160522

(30 points) Part 2:

Find a list of "valid" years, then display them in a "user-friendly" format before taking the inputs from the user. The valid years are years where there is data (i.e., not NaN) in both the MHI and HPI datasets. Continuing from the previous example, the valid years would be 1984-1989, 1991-1999, 2001-2009, and 2011-2021. After your program found the valid years, display it as shown in the figure below:

```
Valid years are 1984-1989, 1991-1999, 2001-2009, 2011-2021.
Enter a valid base year: 1990
Enter a valid base year: 1991
Enter a valid target year: 2010
Enter a valid target year: 2020
```

The figure also shows that, in addition to showing the valid years, the program also takes two inputs from the user: (1) base year; and (2) target year. It also checks for the validity of the inputs and loops (infinitely) to take a new input if the input it receives is invalid. The base year is valid if it is in the list of valid years. The target year is valid if it is in the list of valid years and is different from the base year.

Most of the points in this part are allocated to your program being able to display the list of valid years in a correct format. Please pay attention to how the list is printed, down to the hyphens, commas, spaces, and dots. If, for example, the year 2020 is also missing in either the MHI or HPI dataset, then the message should be "`Valid years are 1984-1989, 1991-1999, 2001-2009, 2011-2019, 2021.`"

If there are less than 2 valid years, then your program should simply terminate without asking for input from the users.

(100 points) Part 3:

After your program receives the base and the target years from the user, it must (1) **plot** a line chart showing a factor change of every data point in the MHI, HPI, and CPI datasets compared to the average value within the base year; and (2) **save** the chart as a PDF file named "nm.pdf". A factor is a multiplier that when multiplied with the initial value gives you the final value. For example, if the MPI values at the years 1991 and 2020 are 26540 and 50906, respectively; then, the factor when the year 2020 (target year) is compared to the year 1991 (base year) is 50906 / 26540 = 1.92.

You should have already noticed that in the HPI and CPI datasets, there are multiple data points within each year. Your program should use the average value of data points within each year as a representative of that year. For example, the HPI dataset has 4 data points: 136.14, 137.58, 138.33, and 140.44, within the year 1991; therefore, 138.12 would represent the value of HPI in the year 1991. When comparing the HPI value between the years 2020 (target) and 1991 (base), the value 357.54, which is an average value of HPI representing the year 2020, is used to compare it with 138.12.

There should be 3 lines representing factor changes of every data point in the 3 datasets: MHI, HPI, and CPI, when comparing it to the average value within the base year. Please note the style in the sample figures below and mimic them. Make sure to use the average value to only represent the base year and use the actual value of every data point to compare with it. For example, the average value for HPI in the year 1991 is 138.12, but there are 4 data points (on different quarters) to plot for HPI in the same year, which are 136.14 / 138.12 = 0.99, 137.58 / 138.12 = 1.00, 138.33 / 138.12 = 1.00, and 140.44 / 138.12 = 1.02, sequentially.

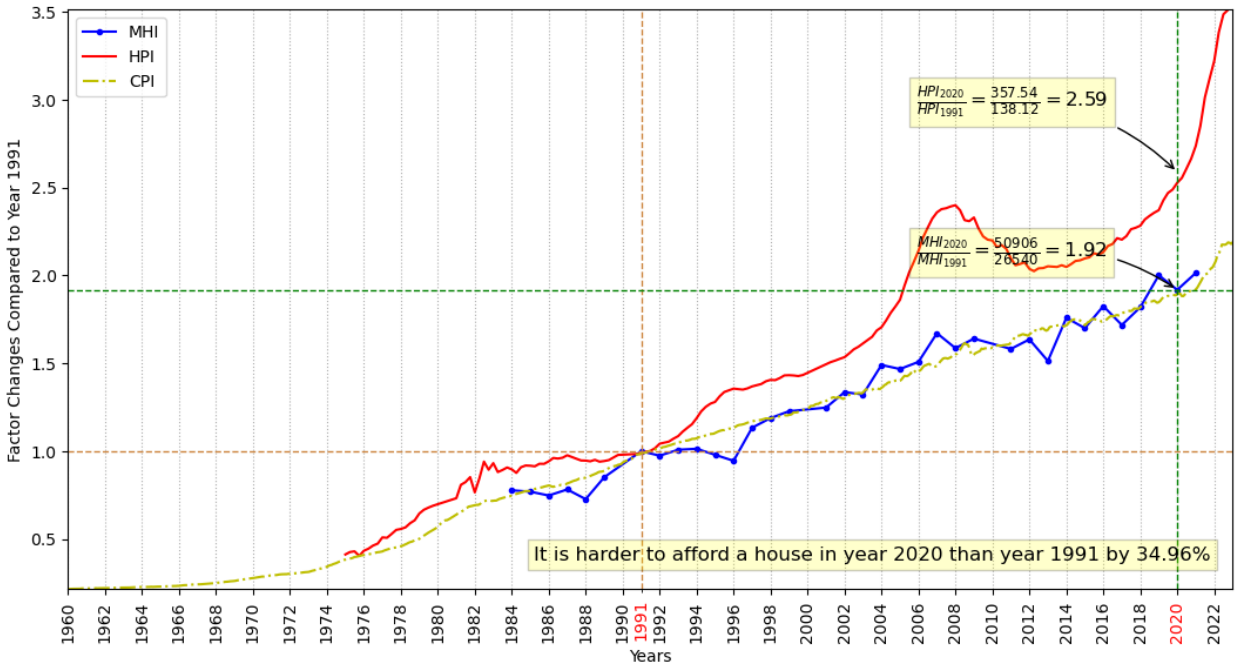
The requirements for the line chart are as follows:

1. The figure size should be set to 12.8 x 6.4 inches with zero margin.
2. The figure saved in the PDF file should only have 0.1 inches padding around the figure.
3. There are labels for the x-axis and y-axis as shown in the sample figures below. Note that the label on the y-axis changes depending on the base year.
4. There is a tick on the x-axis for every even year (e.g., 1960, 1962, 1964, …).
5. There are ticks on the x-axis for the base and the target years.
6. The grid is shown on every tick on the x-axis.
7. There is a legend showing the styles and names of each series of data.
8. There is (1) a brown vertical line on the base year; and (2) a brown horizontal line where the factor change is 1.0. The color I use in the sample figures below is called "peru", actually.
9. There is (1) a green vertical line on the target year; and (2) a green horizontal line where the factor changes of MHI at the target year compared to the base year is.
10. There are 3 text boxes: (1) points to the factor of MHI at the target year; (2) points to the factor of HPI at the target year; and (3) at the bottom-right corner summarizing whether it is harder or easier to afford a house in the target year compared to the base year. The value in

the third box can be determined from the values shown in the first and the second boxes. To receive full points, texts in your chart need to be in the same format as in the sample figures.

11. **[Only for CS-454 students]** The ticks for the base year and the target year should be colored red (or another color if you prefer, but different from the other ticks).

Continuing from the example in Part 2, your program should produce a chart as shown below:



If the base year and the target year are swapped, then the following chart should be produced: